

# HoGent

BEDRIJF  
EN  
ORGANISATIE

## Webapplicaties 1

SASS : **S**yntactically **A**wesome **S**tylesheets

# Inhoud

---

## ▶ CSS

- selectors
- properties met waarden
- cascading regels
- overerving voor het toepassen van de stijlen

## ▶ Problemen

- Weinig gestructureerde code bij gebruik van complexe selectors: descendant, sibling, grouping
- Moeilijk te onderhouden
- Herhaling in waarden (een waarde veranderen moet vaak op meerdere plaatsen): vb kleur, font, ...

# Inhoud

---

## ► Wat als?

- Je variabelen kan definiëren: bijvoorbeeld voor kleuren
    - Je hoeft een hexadecimale kleurwaarde niet langer te memoriseren.
  - Je stijlregels kan hergebruiken
  - Je methodes kan definiëren
  - Je berekeningen kan doen
  - Je stijlregels kan nesten
    - Structuur CSS volgt structuur HTML
- } minder code & overzichtelijker

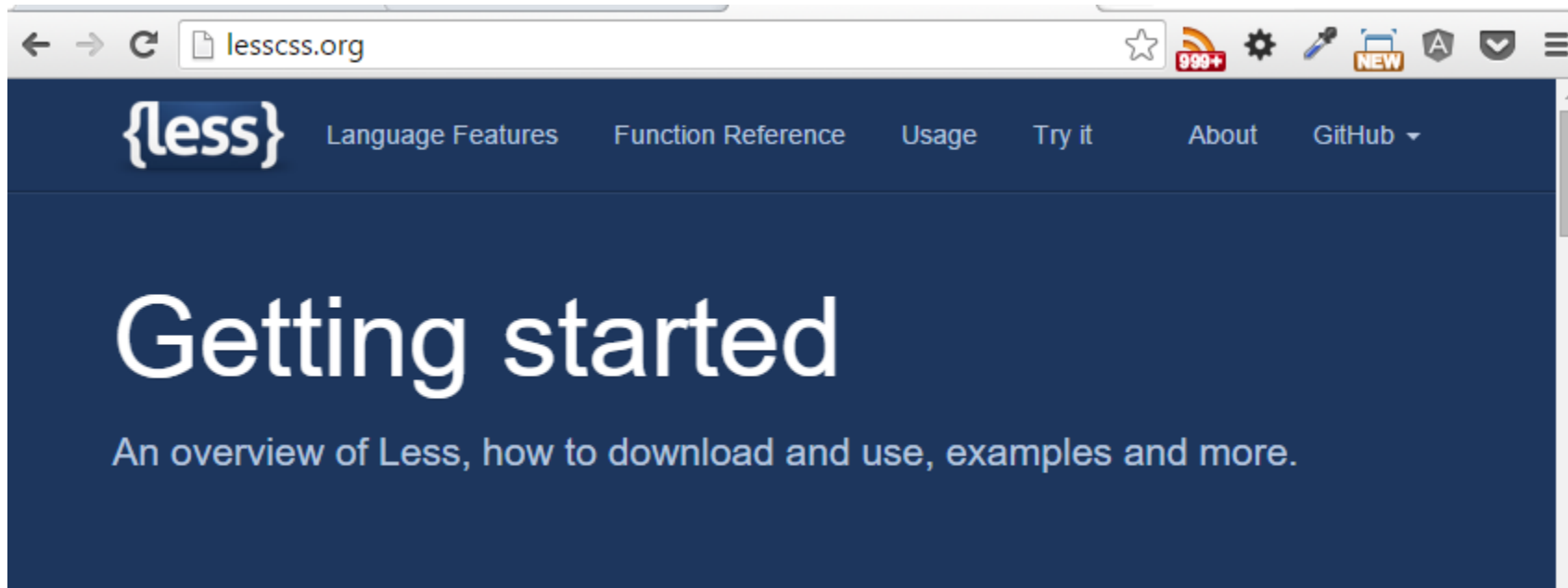
[INSTALL](#)[LEARN SASS](#)[BLOG](#)[DOCUMENTATION](#)[GET INVOLVED](#)[LIBSASS](#)

# CSS with superpowers



Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

Current Release: Selective Steve (3.4.19)

[Release Notes](#)[Fork on Github](#)[Implementation Guide](#)

# SASS

- ▶ Syntactically **A**wesome **S**tylesheets
  - Ontwikkeld in 2007
  - <http://sass-lang.com/>
- ▶ Dient als scripting taal: breidt CSS3 uit met variabelen, math, mixins,...
- ▶ Makkelijker te structureren en dus onderhouden
- ▶ SASS wordt omgezet naar CSS door preprocessor
- ▶ 2 formatteringsconventies
  - SASS: focus op indentaties en shorthand notaties
  - SCSS: gebruikt de css conventies

# SASS

- ▶ .sass: indents en shorthand notaties

```
body #main
  color:#eee
  a.link
    color:#eff
    text-decoration:none|
```

- ▶ .scss: css conventies {} en ;

```
body #main {
  color:#eee;
  a.link {
    color:#eff;
    text-decoration:none;
  }
}
```

# Preprocessing

---

- ▶ SASS/SCSS moet gecompileerd worden naar CSS, die CSS komt op de site (*niet* de scss)
- ▶ Compilatie met behulp van Visual Studio Code extensie "Live Sass Compiler"
  - Installeren via de extension manager in het programma
- ▶ **Vanaf nu geen “gewone” CSS meer schrijven, SCSS is beter!**
- ▶ Automatische compilatie van SCSS
- ▶ Includen van gecompileerde CSS op webpagina

# De belangrijkste features

---

- ▶ Variabelen
- ▶ Nesting
- ▶ Mixins
- ▶ Imports
- ▶ Inheritance
- ▶ Appendix: SassScript



# Variabelen

# Variabelen

**\$variabele: value;**

- ▶ Waarde toekennen aan een variabele
  - Steeds terugkerende waarden krijgen een betekenis
    - Bvb kleurcode #3366cc wordt \$mainFontColor
  - Compilatie zet dit terug om in de correcte CSS schrijfwijze
- ▶ Mogelijke waarden
  - hex values, strings, kleuren, nummers, booleans, en zelfs lijsten van waarden
- ▶ Variabelen hebben een scope
  - Gedefinieerd binnen selector -> enkel gekend binnen die selector
  - Anders globale scope

# Variabelen - voorbeelden

## ▶ **Strings**

- `$myString: "your text here";`

## ▶ **Numbers**

- `$myNum: 10px;`
- Alle getallen, px, em, %

## ▶ **Colors**

- `$myColor: white;`
- Hexadecimale waarden, rgba, hsla

## ▶ **Booleans**

- `$myBool: true;`

## ▶ **Lists**

- `$myItemList: 1px solid red;`

## ▶ **Nulls**

- `$myVar: null;`

# Variabelen

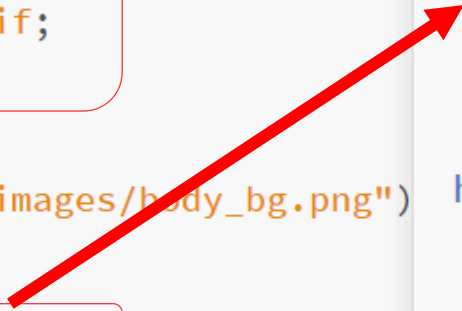
- ▶ Gebruik variabelen voor terugkerende waarden
  - fontinstellingen, kleurenpalet,...
- ▶ Voorbeeld definitie  
`$red:#ed1c24;`
- ▶ Voorbeeld gebruik  
`body{ background-color:$red;}`  
`p {color: $red}`
- ▶ Definieer variabelen bovenaan in .scss
  - Variabelen kan je pas gebruiken eens ze gedefinieerd zijn (logisch)

**.SCSS**

```
/*variabelen*/
$red:#ed1c24;
$gray:#424242;
$font:'Roboto', sans-serif;
$basefontsize : 1em;
/*algemeen*/
body {
    background: url("../images/body_bg.png");
    color: $gray;
    font-family: $font;
    font-size: $basefontsize ;
    line-height: 1.5;
}
h2 {font-size: 2em;}
a {
    color: $red;
    text-decoration: none;
}
a:visited { color: $gray;}
a:hover,
a:visited {
    text-decoration: underline;
}
p { margin: 0.5em 0;}
```

**.CSS**

```
/*variabelen*/
/*algemeen*/
body {
    background: url("../images/body_bg.
    color: #424242;
    font-family: "Roboto", sans-serif;
    font-size: 1em;
    line-height: 1.5; }
h2 {
    font-size: 2em; }
a {
    color: #ed1c24;
    text-decoration: none; }
a:visited {
    color: #424242; }
a:hover,
a:visited {
    text-decoration: underline; }
p {
    margin: 0.5em 0; }
```



# Variabelen

## ► String interpolatie

- De waarde van een variabele gebruiken als onderdeel van een stukje tekst (url, naam van een property of selector...)

definitie: `$variabele: value;`

gebruik: `#{$naamVariabele}`

- Voorbeeld:

- Een variabele met pad naar de image folder

```
$imagesFolder : "../images/";
```

- Gebruik:

```
body {  
  background: url("#{ $imagesFolder }body_bg.png") repeat top left;
```

# Nesting

# Nesting

- ▶ Een oplossing voor de eindeloos lange selectors in css
  - Het nesten van stijlregels
  - Structuur van HTML komt zo ook terug in SCSS
  - Voorbeeld : de header opmaken

**.SCSS**

```
header {  
  overflow: hidden;  
  padding: 20px 0 0 0;  
  h1 {  
    color: $red;  
    float: left;  
    font-size: 1.5em;  
    line-height: 1.2;  
    margin-top: 15px;  
    width: 350px;  
  }  
}
```

**.CSS**

```
header {  
  overflow: hidden;  
  padding: 20px 0 0 0;  
}  
  
header h1 {  
  color: #ed1c24;  
  float: left;  
  font-size: 1.5em;  
  line-height: 1.2;  
  margin-top: 15px;  
  width: 350px;  
}
```



# Nesting

- Nesten van stijlregels gebaseerd op een relatie (>, ~, +, ...)

.SCSS

```
header {  
  overflow: hidden;  
  padding: 20px 0 0 0;  
  h1 {  
    color: $red;  
    float: left;  
    font-size: 1.5em;  
    line-height: 1.2;  
    margin-top: 15px;  
    width: 350px;  
  }  
  > a > img {  
    float: left;  
    height: 118px;  
    width: 142px;  
    margin-bottom: 5px;  
  }  
}
```

.CSS

```
header {  
  overflow: hidden;  
  padding: 20px 0 0 0;  
}  
  
header h1 {  
  color: #ed1c24;  
  float: left;  
  font-size: 1.5em;  
  line-height: 1.2;  
  margin-top: 15px;  
  width: 350px;  
}  
  
header > a > img {  
  float: left;  
  height: 118px;  
  width: 142px;  
}
```

# Nesting

Opm : Je kan natuurlijk ook kiezen om de media queries niet te nesten.

- ▶ Media bubbling: properties voor specifieke media tussenin plaatsen

.scss

```
header {
  overflow: hidden;
  padding: 20px 0 0 0;
  h1 {
    color: $red;
    float: left;
    font-size: 1.5em;
    line-height: 1.2;
    margin-top: 15px;
    width: 350px;
    @media screen and (min-width: 992px) {
      font-size: 2em;
      margin-top: 30px;
      width: 80%;
    }
  }
}
> a > img {
  float: left;
  height: 118px;
```

.css

```
header {
  overflow: hidden;
  padding: 20px 0 0 0;
}

header h1 {
  color: #ed1c24;
  float: left;
  font-size: 1.5em;
  line-height: 1.2;
  margin-top: 15px;
  width: 350px;
}

@media screen
  and (min-width: 992px) {
  header h1 {
    font-size: 2em;
    margin-top: 30px;
    width: 80%;
  }
}
```

# Nesting

- ▶ & : verwijst naar de parent selector
  - Vooral nodig voor pseudoclasses

.SCSS

```
a {  
  color: $red;  
  text-decoration: none;  
  &:visited {  
    color: $gray;  
  }  
  &:hover,  
  &:visited {  
    text-decoration: underline;  
  }  
}
```

.CSS

```
a {  
  color: #ed1c24;  
  text-decoration: none;  
}  
  
a:visited {  
  color: #424242;  
}  
  
a:hover,  
a:visited {  
  text-decoration: underline;  
}
```

# Nesting

## ► Nested properties

**.SCSS**

```
body {  
  background: url("#{imagesFolder}body_bg.png") repeat top left;  
  color: $gray;  
  font:{  
    family: $font;  
    size: $basefontsize;  
  }  
  line-height: 1.5;  
}
```

**.CSS**

```
body {  
  background: url("../images/body_bg.png") repeat top left;  
  color: #424242;  
  font-family: "Roboto", sans-serif;  
  font-size: 1em;  
  line-height: 1.5;  
}
```

# Nesting - voorbeeld

## .SCSS

```
/*Tabellen*/  
table,  
th,  
td {  
    border: 1px solid $gray;  
    padding: 0.5em;  
}  
  
th {  
    background-color: $gray;  
    color: white;  
}  
  
td table {  
    &,  
    th,  
    td {  
        border: none;  
    }  
}
```

## .CSS

```
/*Tabellen*/  
  
table,  
th,  
td {  
    border: 1px solid #424242;  
    padding: 0.5em;  
}  
  
td table,  
td table th,  
td table td {  
    border: none;  
}  
  
th {  
    background-color: #424242;  
    color: white;  
}
```

# Mixins

# Mixins

- ▶ Mixins laten toe om verzamelingen eigenschappen één keer te definiëren en ze dan te hergebruiken doorheen de rest van de CSS code

.scss

```
@mixin AAGentStijl {  
  background: ■ #054B91;  
  border-radius: 5px;  
  color: □ white;  
}  
  
a {  
  @include AAGentStijl;  
}  
  
p {  
  @include AAGentStijl;  
}
```

Hogent

.css

```
a {  
  background: ■ #054B91;  
  border-radius: 5px;  
  color: □ white;  
}  
  
p {  
  background: ■ #054B91;  
  border-radius: 5px;  
  color: □ white;  
}
```

# Mixins

- ▶ Kan ook in combinatie met parameters

**.SCSS**

```
@mixin AAGentStijl($roundingPX) {  
  background: ■ #054B91;  
  border-radius: $roundingPX;  
  color: □ white;  
}  
  
a {  
  @include AAGentStijl(5px);  
}  
  
p {  
  @include AAGentStijl(10px);  
}
```

**.CSS**

```
a {  
  background: ■ #054B91;  
  border-radius: 5px;  
  color: □ white;  
}  
  
p {  
  background: ■ #054B91;  
  border-radius: 10px;  
  color: □ white;  
}
```



# Mixins

- ▶ Default waarden en benoemde parameters
  - Benoemde parameter : je geeft de naam + de waarde van de parameter op bij aanroep mixin, voor de overige parameters wordt default gebruikt
  - Voorbeeld : mixin voor border

.SCSS

```
@mixin border($size:1px, $color:black) {  
    border:$size solid $color;  
}
```

```
nav {  
    @include border();  
}
```

```
table,  
th,  
td {  
    @include border($color:$gray);  
    padding: 0.5em;  
}
```

.CSS

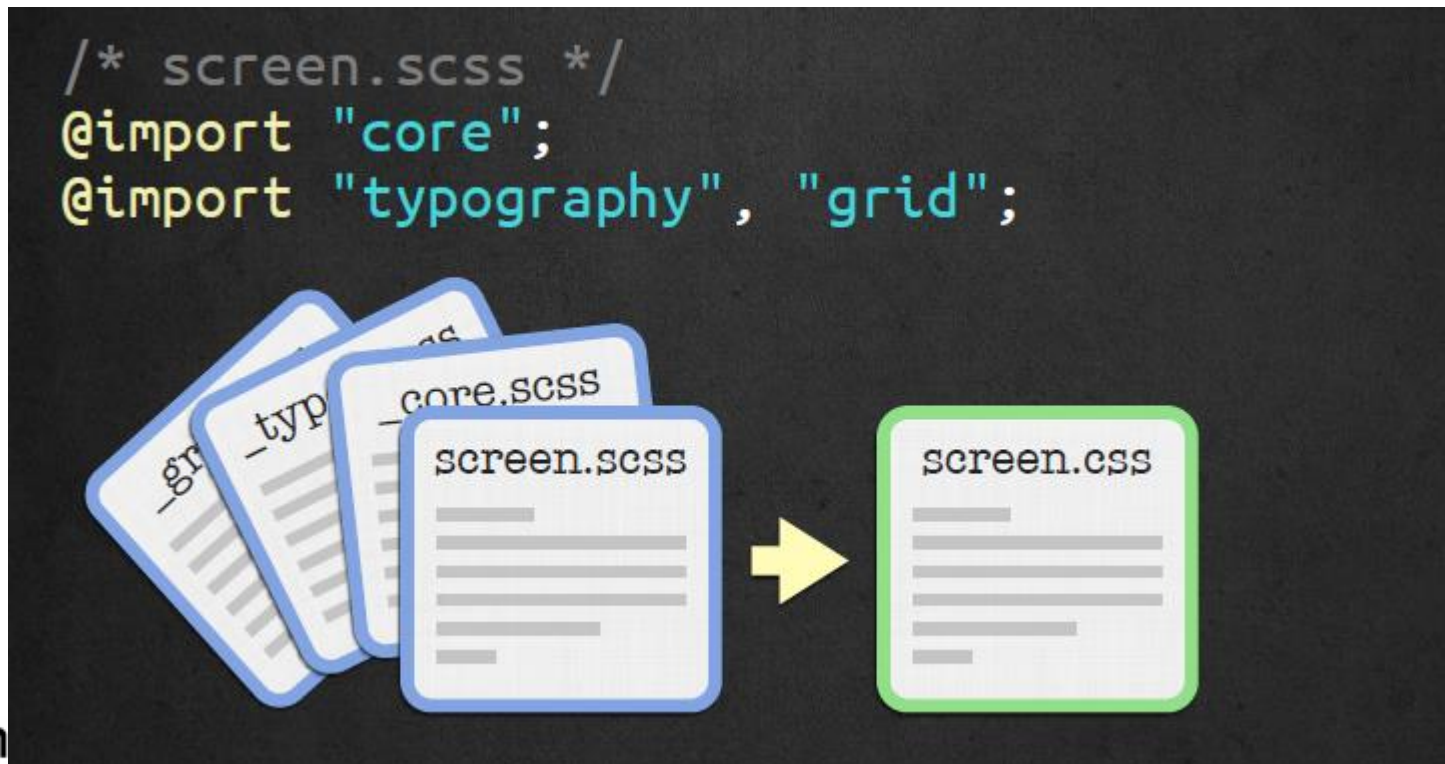
```
nav {  
    border: 1px solid black;  
}
```

```
table, th, td {  
    border: 1px solid #424242;  
    padding: 0.5em;  
}
```

# Partials

# Partials

- ▶ Je kan **partial** bestanden creëren om de code op te delen.
- ▶ Gebruik `@import` om bestand te importeren
- ▶ Partials : Begin met `_`



# Partials

- ▶ Maak partials aan voor de variabelen en voor de mixins.

CSS

`_mixins.scss`

`_variables.scss`

*`_variables.scss`*

```
1  /*variabelen*/
2
3  $red:#ed1c24;
4  $gray:#424242;
5  $font:'Roboto',
6  sans-serif;
7  $basefontsize: 1em;
8  $imagesFolder: "../images/";
```

*`site.scss`*

```
@import "variables";
@import "mixins";
```

# Inheritance

# Inheritance

- ▶ @extend laat je toe stijlregels van een andere selector over te nemen
  - Zonder sass : combineren van 2 classes

- css

```
/*formulier */  
.button {  
    background: #424242;  
    color:white;  
    padding:0.2em 0.8em;  
    border-radius:0.4em;  
}  
  
.button-submit {  
    background: #ed1c24;  
}
```

- html

```
<input type="submit" value="verzend"  
       class="button button-submit"/>
```

# Inheritance

## ► Met Sass : @extend

**.SCSS**

```
.button {  
  background: $gray;  
  color: white;  
  padding: 0.2em 0.8em;  
  border-radius: 0.4em;  
}  
  
.button-submit {  
  @extend .button;  
  background: $red;  
}
```

**.CSS**

```
/*formulier */  
.button, .button-submit {  
  background: #424242;  
  color: white;  
  padding: 0.2em 0.8em;  
  border-radius: 0.4em; }  
  
.button-submit {  
  background: #ed1c24; }
```

**.html**

```
<input type="submit" value="verzend" class="button-submit"/>
```

# Appendix : SassScript



# SassScript

---

- ▶ Naast de CSS property syntax, bevat Sass ook een extensie genaamd SassScript. SassScript laat toe om in properties variabelen, rekenkundige operatoren en extra functies te gebruiken.
- ▶ Dit is enkel een kleine greep uit het aanbod
- ▶ Meer op : [http://sass-lang.com/documentation/file.SASS\\_REFERENCE.html](http://sass-lang.com/documentation/file.SASS_REFERENCE.html)

# SassScript : Data Types

## ▶ Data types

- number : e.g. 1.2, 13, 10px
- string met of zonder quotes : "foo", 'bar', baz
- color : blue, #04a3f9, rgba(255, 0, 0, 0.5)
- boolean : true, false
- Null
- Lijst : reeks van waarden gescheiden door komma of spatie
  - 1.5em 1em 0 2em
  - Helvetica, Arial, sans-serif
- Map : key-value paren
  - (key1: value1, key2: value2)

# SassScript : Data types & functies

- ▶ Op de data types zijn tal van operatoren gedefinieerd
- ▶ Number functions
  - `percentage(13/25)` // 52%
  - `round(2.4)` // 2
  - `ceil(2.2)` // 3
  - `floor(2.6)` // 2
  - `abs(-24)` // 24

# SassScript : Data types & functies

- ▶ Functies gedefinieerd op color



# SassScript : Operatoren

- ▶ Math operators (+, -, \*, /, %)
  - #sidebar{ fontsize :1em + 1em;} //2em
  - #sidebar{ fontsize:1em – 1em;} //0em
  - #sidebar{ fontsize:6px \* 4;} //24px
  - #sidebar{ fontsize:18px % 5;} //3px
  - \$container: 960px;  
\$main:680px;  
#sidebar{ width:\$container - \$main;} //310px

# SassScript : Operatoren

- ▶ Relational operators (<, >, <=, >=) evaluate numbers
  - `1 < 20 // true`
  - `10 <= 20 // true`
  - `4 > 1 // true`
  - `4 >= 1 // true`
- ▶ Comparison operators (==, !=) evaluate all data types
  - `1 + 1 == 2 // true`
  - `small != big // true`
  - `#000 == black // true`

# SassScript : control directives

- ▶ if()
  - ▶ @if
  - ▶ @while
  - ▶ @for
  - ▶ @each
- 
- ▶ Voorbeelden zie [http://sass-lang.com/documentation/file.SASS\\_REFERENCE.html#control\\_directives\\_expressions](http://sass-lang.com/documentation/file.SASS_REFERENCE.html#control_directives_expressions)

# SassScript : control directives

- ▶ Een voorbeeld : @while

SCSS

```
$level: 0;
@while $level < 5 {
  .tag-#{ $level + 1 } {
    font-size: .7em + ($level * .5em);
  }
  $level: $level + 1;
}
```

CSS

```
.tag-1 { font-size: 0.7em; }
.tag-2 { font-size: 1.2em; }
.tag-3 { font-size: 1.7em; }
.tag-4 { font-size: 2.2em; }
.tag-5 { font-size: 2.7em; }
```



# **Appendix : Referenties**

# Referenties en interessante links

---

- ▶ Sass documentatie :

<http://sass-lang.com/documentation/>

- ▶ Sass playground :

<http://sassmeister.com/>